Detecting Word Usage Errors in Chinese Sentences for Learning Chinese as a Foreign Language

Yow-Ting Shiue and Hsin-Hsi Chen

Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan
orina1123@gmail.com; hhchen@ntu.edu.tw

ABSTARCT

Recently more and more people are learning Chinese, and an automated error detection system can be helpful for the learners. This paper proposes n-gram features, dependency count features, dependency bigram features, and single-character features to determine if a Chinese sentence contains word usage errors, in which a word is written as a wrong form or the word selection is inappropriate. Experiments on the HSK corpus show that the classifier combining all sets of features achieves an accuracy of 0.8423. The best precision we achieve is 0.9536, indicating that our system is reliable and seldom produces misleading results.

Introduction

- The flexibility of the Chinese language makes error detection more challenging than other languages.
- According to the analysis on the HSK dynamic composition corpus created by Beijing Language and Culture University, word usage error (WUE) with error tag CC, is the most frequent type of error at the lexical level.
- Four major subtypes of CC error defined in the HSK corpus: (misused form, correct form).
  1) Character disorder in a word, e.g., (先首, 首先) (first of all) and (眾所知周, 眾所周知) (as we all know).
  2) Incorrect selection of a word, e.g., 雖然現在還沒有(實踐, 實現), … (while it is not yet implemented, …).
  3) Non-existent word, e.g., (農作品, 農產品) (agricultural product).
  4) Word collocation error, e.g., 最好的辦法是兩個都(走去, 保持)平衡 (The best way is to keep both balance).
- This paper:
  - Morphological error (W): CC (1) and (3)
  - Usage error (U): CC (2) and (4)

Data Preparation

- Both wrong and correct sentences are selected from the HSK corpus.
- To simplify the problem, we convert a sentence with n errors into n sentences, each of which with only one error.
- In Chinese, a sentence is usually composed of several segments separated by comma " ， ". We consider a segment as a unit of WUE detection.
- ICTCLAS Chinese Word Segmentation System
- Length(segment) = # words in the segmentation result.
- Table 1 shows that learners make usage errors more often than writing a word as a wrong form.
- Balanced dataset: randomly select 15,000 correct and WUE segments

	W Error	U Error
HSK WUE	(1) & (3)	(2) & (4)
#segments	4,010	13,314

Table 1: Distribution of WUEs

WUE Detection

Google n-gram features (G)

- Chinese version of Google Web 5-gram
- For every word sequence of length n (n=2, 3, 4, 5), we calculate the n-gram probability by Maximum Likelihood Estimation. Taking trigram for example, the probability is:

$$p(w_i|w_{i-2},w_{i-1}) = \frac{c(w_{i-2},w_{i-1},w_i)}{c(w_{i-2},w_{i-1})} \qquad (1)$$

- All n-gram features are concatenated into a feature vector G = (g2, g3, g4, g5), where

$$g_n = \sum_{i=n}^{L} p(w_i|w_{i-n+1},\dots,w_{i-1}) \qquad (2)$$

WUE Detection (cont.)

Dependency count feature (D)

- Errors in a sentence affect the result of segmentation and parsing.
- We take the count of each type of dependency of Stanford Parser output as a set of features.
- Two types of "count" for each dependency:
  1) Internal count: counts the occurrence if the two words are both in the target segment
  2) External count: counts as long as one of the words is in the target segment.

Dependency bigram feature (B)

- Long distance dependency is common in Chinese sentences.
- Example:
  親身/體會/了/一場/永遠/難忘/的/電單車/意外
- Dependencies: nsubj(體會-2, 親身-1), dobj(體會-2, 意外-9), …
- Compose the two words in each dependency:
  (親身, 體會), (體會, 意外)…
  → query the Google n-gram corpus
  → calculate the bigram probabilities
- Sum the bigram probabilities of each type (internal/external).

Single character feature (S)

- A non-existent Chinese word (W-type error) is usually separated into several single-character words after segmentation
- Define the following features:
  1) # contiguous single-character blocks
  2) # contiguous single-character blocks with length no less than 2
  3) Length of the maximum contiguous single-character block
  4) Sum of the lengths of all contiguous single-character blocks

Experimental Results and Analysis

	Model: support vector machine				Model: decision tree			
Feature	Acc.	Prec.	Recall	F1	Acc.	Prec.	Recall	F1
G	0.7706	0.7650	**0.7813**	0.7731	**0.8333**	0.9532	0.7011	0.8079
D	0.6586	0.6771	0.6068	0.6400	0.6242	0.6248	0.6228	0.6238
B	0.6102	0.6226	0.5595	0.5894	0.6148	0.6094	0.6447	0.6266
S	0.6217	0.6435	0.5456	0.5905	0.6196	0.6453	0.5314	0.5828
DB	0.6534	0.6702	0.6041	0.6354	0.6231	0.6272	0.6114	0.6192
GD	0.7638	0.7710	0.7507	0.7607	**0.8325**	0.9513	0.7009	0.8071
GB	0.7550	0.7453	0.7749	0.7598	**0.8316**	**0.9536**	0.6972	0.8055
GS	**0.7858**	**0.7885**	0.7810	**0.7874**	**0.8341**	0.9503	**0.7050**	**0.8095**
GDBS	0.7716	0.7765	0.7628	0.7696	**0.8332**	0.9486	0.7046	0.8086

Table 2: Performance of SVM and decision tree

	Model: random forest			
Feature	Acc.	Prec.	Recall	F1
G	0.8324	**0.9496**	0.7021	0.8073
GD	0.8371	0.9023	0.7560	0.8227
GB	0.8386	0.9251	0.7369	0.8203
GS	0.8391	0.9443	0.7206	0.8174
GDBS	**0.8423**	0.8998	**0.7705**	**0.8301**

Table 3: Performance of random forest

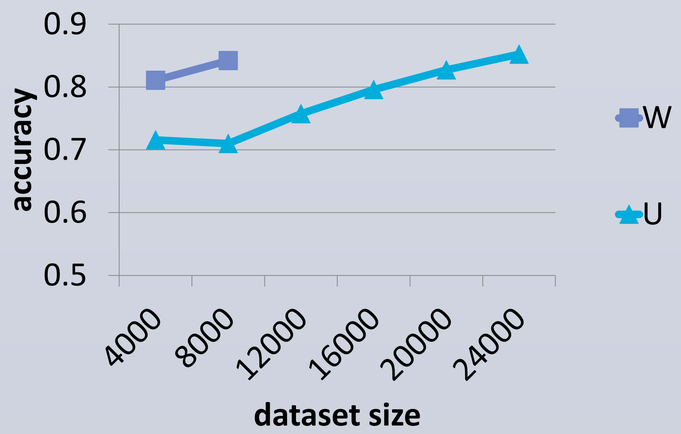

Figure 1: Accuracy vs. dataset size